Physical Complexity of Variable Length Symbolic Sequences

Gerard Briscoe[a], Philippe De Wilde[b]

[a] *Systems Research Group, Computer Laboratory, University of Cambridge, United Kingdom, gerard.briscoe@cl.cam.ac.uk*
[b] *Intelligent Systems Lab, Department of Computer Science, Heriot Watt University, United Kingdom, p.de_wilde@hw.ac.uk*

Abstract

A measure called Physical Complexity is established and calculated for a population of sequences, based on statistical physics, automata theory, and information theory. It is a measure of the quantity of information in an organism's genome. It is based on Shannon's entropy, measuring the information in a population evolved in its environment, by using entropy to estimate the randomness in the genome. It is calculated from the difference between the maximal entropy of the population and the actual entropy of the population when in its environment, estimated by counting the number of fixed *loci* in the sequences of a population. Up to now, Physical Complexity has only been formulated for populations of sequences with the same length. Here, we investigate an extension to support *variable* length populations. We then build upon this to construct a measure for the efficiency of information storage, which we later use in understanding clustering within populations. Finally, we investigate our extended Physical Complexity through simulations, showing it to be consistent with the original.

Key words: complexity, entropy, clustering, evolution, population

1. Introduction

Physical Complexity was born [1] from the need to determine the proportion of information in sequences of DNA, because it has long been established [28] that the information contained is not directly proportional to the length, known as the C-value enigma/paradox [15]. However, because Physical Complexity analyses an ensemble of DNA sequences, the consistency between the different solutions shows the information, and the differences the redundancy [2]. Entropy, a measure of disorder [30], is used to determine the redundancy from the information in the ensemble (populations). Physical Complexity therefore provides a context-relative definition for the complexity of a population without needing to define the context (environment) explicitly [3]. It is widely recognized that complexity is best measured using thermodynamic depth. Lloyd and Pagels make this case in [23]. The introduction of thermodynamic depth allows them to prove "the average complexity of a state must be proportional to the Shannon entropy of the set of trajectories that experiment determines can lead to that state". They further relate thermodynamic depth to computational complexity, and derive a lower bound from the mutual information. We follow a slightly different route, following [3, 32], but the Shannon entropy also features in our complexity definition (4).

Physical Complexity is currently formulated for a population of *same length* symbolic sequences [3]. So, we decided to extend Physical Complexity to include populations of *variable length* symbolic sequences, and therefore allow for its improved and wider applicability [8, 7, 10].

2. Physical Complexity

Physical Complexity was derived [3] from the notion of *conditional complexity* defined by Kolmogorov, which is different from traditional Kolmogorov complexity and states that the determination of complexity of a sequence is conditional on the environment in which the sequence is interpreted [22]. In contrast, traditional Kolmogorov-Chaitin (KC) complexity is only conditional on the implicit rules of mathematics necessary to interpret a programme on the tape of a Turing Machine (TM), and nothing else [22]. So, if we consider a TM that takes a tape e as input (which represents its physical environment), including the particular rules of mathematics of this *world*; without such a tape, this TM is incapable of computing anything, except for writing to the output what it reads in the input. Thus, without tape e all sequences s have maximal KC-complexity, because there is nothing by which to determine regularity [3]. However, *conditional complexity* can be stated as the length of the smallest programme that computes sequence s from an environment e,

$$K(s|e) = \min\{|p| : s = C_T(p, e)\}, \tag{1}$$

where $C_T(p, e)$ denotes the result of running programme p on Turing Machine T with the input sequence e [3]. This is not yet Physical Complexity, but rather, it is the smallest programme that computes the sequence s from an environment e, in the limit of sequences of infinite length, containing only the bits that are entirely unrelated to e, since, if they were not, they could be obtained from e with a programme of a size tending to zero [3]. The Physical Complexity $K(s : e)$ can now be defined as the number of bits that are meaningful in sequence s (that can be obtained from e with a programme of vanishing size), and is given by the *mutual complexity* [19],

$$K(s:e) = K(s|\emptyset) - K(s|e), \tag{2}$$

where $K(s|\emptyset)$ is the unconditional complexity with an empty input tape, $e \equiv \emptyset$ [3]. This is different from the Kolmogorov complexity, because in Kolmogorov's construction the rules of mathematics were given to the TM [22]. As argued above, every sequence s is random if no environment e is specified, as non-randomness can only exist for a specific world or environment. Thus, $K(s|\emptyset)$ is always maximal,

$$K(s|\emptyset) = |s|, \tag{3}$$

and is given by the length of s [3]. So (2) represents the length of the sequence s, minus those bits that cannot be obtained from e. So, conversely (2) represents the number of bits that can be obtained in a sequence s, by a computation with vanishing programme size, from e. Thus, $K(s:e)$ represents the Physical Complexity of s [3]. The determination of the Physical Complexity, $K(s:e)$, of a sequence s with a description of the environment e is not practical. Meaning that it cannot generally be determined by inspection, because its impossible to determine which, and how many, of the bits of sequence s correspond to information about the environment e. The reason is that we are generally unaware of the coding used to code information about e in s, and therefore coding and non-coding bits look entirely alike [3]. However, it is possible to distinguish coding from non-coding bits if we are given multiple copies of sequences that have adapted to the environment, or more generally, if a statistical ensemble (population) of sequences is available to us. Then, coding bits are revealed by non-uniform probability distributions across the population (*conserved sites*), whereas random bits have uniform distributions (*volatile sites*) [3]. The determination of complexity then becomes an exercise in information theory, because the average complexity $\langle K \rangle$, in the limit of infinitely long strings, tends to the entropy of the ensemble of strings S[1] [32],

$$\langle K(s) \rangle_S = \sum_{s \in S} p(s) K(s) \approx H(S), \tag{4}$$

where H is defined from Shannon's (information) entropy [24], and is given by

$$H(S) = log_n(S), \tag{5}$$

where n is the number of symbols available for encoding. If each symbol is equally probable, we can rewrite the above function as

$$\begin{aligned} H(S) &= -log_n(1/S) \\ &= -log_n(p), \end{aligned} \tag{6}$$

where p is the probability of occurrence of any one of the symbols. For a source that outputs an infinite sequence of bits, to communicate a finite set of symbols S, Shannon generalised the above function to express an average symbol length [24]. This derivation is easier to see for a large, but finite, number of symbols N,

$$\begin{aligned} H(S) &= \frac{\sum_{i=1}^{S} N_i \left[-\log_N(1/S_i)\right]}{\sum_{i=1}^{S} N_i} = \frac{\sum_{i=1}^{S} N_i \left[-\log_N(1/S_i)\right]}{N} \\ &= -\sum_{i=1}^{S} \frac{N_i}{N} \left[\log_N(1/S_i)\right] = -\sum_{i=1}^{S} p_i \log_N(p_i), \end{aligned} \tag{7}$$

where N_i is the number of occurrences of the symbol S_i. So, given (4) and (7), the *average complexity* of the sequences s of a population S, $\langle K(s) \rangle_S$, tends to the entropy of the sequences s in the ensemble S [3],

$$\langle K(s) \rangle_S = -\sum_{s \in S} p(s) \log p(s). \tag{8}$$

(8) remains consistent with (3) as the determination of $K(s|\emptyset)$, sequence s without an environment e, must equal the sequence's length $|s|$, because Shannon's formula for entropy is an average logarithmic measure of the symbol sets [24], and so the maximum entropy of a population is equivalent to the length of the sequences in the population, $H_{max}(S) = |s|$. Indeed, if nothing is known about the environment to which a sequence s pertains, then according to the *principle of indifference*[2], the probability distribution $p(s)$ must be uniformly random. However, if an environment e is given we have some information about the system, and the probability distribution will be non-uniform. Indeed, it can be shown that for every probability distribution $p(s|e)$, to find sequence s given environment e, we have

$$H(S|e) \leq H(S|\emptyset) = |s|, \tag{9}$$

because of the concavity of Shannon entropy [3]. So, the difference between the maximal entropy $H(S|\emptyset) = |s|$ and $H(S|e)$, according to the construction outlined above, represents the average number of bits in sequence s taken from the population S that can be obtained by zero-length universal programmes from the environment e. Therefore, the average mutual complexity of sequences s in a population S, given an environment e, is

$$\begin{aligned} \langle K(s:e) \rangle_S &= \sum_{s \in S} p(s) K(s:e) \\ &\approx H(S|\emptyset) - H(S|e) \\ &\equiv I(S|e), \end{aligned} \tag{10}$$

where $I(S|e)$ is the information about the environment e stored in the population S, which we identify with the Physical Complexity [3]. To estimate $I(S|e)$ it is necessary to estimate the entropy $H(S|e)$ using a representative population of sequences

[1] This holds for near-optimal codings. For strings s that do not code perfectly we have $\langle K \rangle \geq H$ [31].

[2] The *principle of indifference* states that if there are $n > 1$ mutually exclusive and collectively exhaustive possibilities, which are indistinguishable except for their names then each possibility should be assigned an equal probability $\frac{1}{n}$ [17].

S for a given environment e, by summing, over the sequences s of the population S, the probability $p(s|e)$ multiplied by the logarithm of the probability $p(s|e)$,

$$H(S|e) = -\sum_{s\in S} p(s|e)\log p(s|e). \tag{11}$$

The entropy $H(S|e)$ can be estimated by summing the per-site $H(i)$ entropies of the sequence,

$$H(S|e) \approx \sum_{i=1}^{|s|} H(i), \tag{12}$$

where i is a site in the sequence s [3]. Random sites are identified by a nearly uniform probability distribution, and contribute positively to the entropy, whereas non-random sites (which have strongly peaked distributions) contribute very little [3]. So, the Physical Complexity, the average mutual complexity of sequences s in a population S for an environment e, $\langle K(s:e)\rangle_S$, abbreviated as C, is the maximal entropy $H(S|\emptyset)$ minus the sum of the per-site entropies,

$$C = H(S|\emptyset) - \sum_{i=1}^{|s|} H(i). \tag{13}$$

If the sequences s are constructed from an alphabet, a set D, then the per site entropy $H(i)$ for the sequences is

$$H(i) = -\sum_{d\in D} p_d(i)\log_{|D|} p_d(i), \tag{14}$$

where i is a site in the sequences ranging between one and the length of the sequences ℓ, D is the alphabet of characters found in the sequences, and $p_d(i)$ is the probability that site i (in the sequences) takes on character d from the alphabet D, with the sum of the $p_d(i)$ probabilities for each site i equalling one, $\sum_{d\in D} p_d(i) = 1$ [3]. Taking the log to the base $|D|$ conveniently normalises $H(i)$ to range between zero and one,

$$0 \leq H(i) \leq 1. \tag{15}$$

If the site i is identical across the population it will have no entropy,

$$H_{\min}(i) = 0. \tag{16}$$

If the content of site i is uniformly random, i.e. the $p_d(i)$ probabilities all equal to $\frac{1}{|D|}$, it will have maximum entropy,

$$H_{\max}(i) = 1. \tag{17}$$

When the entropy of $H(i)$ is at its minimum of zero, then the site i holds information, as every sample shows the same character of the alphabet. When the entropy of $H(i)$ is at its maximum of one, the character found in the site i is uniformly random and therefore holds no information. So, the amount of information is the maximal entropy of the site (17) minus the actual per-site entropy (14) [3],

$$\begin{aligned} I(i) &= H_{\max}(i) - H(i) \\ &= 1 - H(i). \end{aligned} \tag{18}$$

For clarity the length of the sequences $|s|$ will be abbreviated to ℓ [3],

$$|s| \equiv \ell. \tag{19}$$

So, the complexity of a population S, of sequences s, is the maximal entropy of the population (equivalent to the length of the sequences) ℓ, minus the sum, over the length ℓ, of the per-site entropies $H(i)$,

$$C = \ell - \sum_{i=1}^{\ell} H(i), \tag{20}$$

given (13), (9) and (19) [3]. The equivalence of the maximum complexity to the length matches the intuitive understanding that if a population of sequences of length ℓ has no redundancy, then their complexity is their length ℓ.

If G represents the set of all possible genotypes constructed from an alphabet D that are of length ℓ, then the size (cardinality) of $|G|$ is equal to the size of the alphabet $|D|$ raised to the length ℓ,

$$|G| = |D|^{\ell}. \tag{21}$$

For the complexity measure to be accurate, a sample size of $|D|^{\ell}$ is suggested to minimise the error [3, 5], but such a large quantity can be computationally infeasible. The definition's creator, for practical applications, chooses a population size of $|D|\ell$, which is sufficient to show any trends present. So, for a population of sequences S we choose, with the definition's creator, a computationally feasible population size of $|D|$ times ℓ,

$$|S| \;\geq\; |D|\ell. \tag{22}$$

The size of the alphabet, $|D|$, depends on the domain to which Physical Complexity is applied. For example, the alphabet of RNA is its four nucleotides, $D = \{A, C, G, U\}$, and therefore $|D| = 4$ [3].

Our complexity measure is inexorably dependent on Shannon entropy, because of the results in [32] that lead to equation (4). The symbolic sequences that we analyse are encoding computer programmes that are evolved using evolutionary dynamics [8, 7, 10, 13, 12, 9, 11]. We have no a-priori knowledge about the statistics of the sequences. If we were analyzing real DNA or financial time series, we would have knowledge of nonstationarity on different length scales, Lévy distributions, or other features of non-well behaved systems. If that was the case, it would be appropriate to consider Jensen-Shannon divergence in its regular [16] or non-logarithmic version [20]. The latter uses Tsallis entropy [29]. This has had impressive applications in financial time series [26, 27, 25]. Gell-Mann and Lloyd [14] also point out applications in systems on the edge of chaos. If we could prove that our strings encoding computer programmes (multi-agent systems) had similar abnormal statistics, we would use measures based on non-logarithmic entropy. However, in the absence of such information, we will follow Adami [3] in his use of [32] and use Shannon entropy.

3. Variable Length Sequences

Physical Complexity is currently formulated for a population of sequences of the same length [3], and so we will now investigate an extension to include populations of variable length sequences. This will require changing and re-justifying the fundamental assumptions, specifically the conditions and limits upon which Physical Complexity operates. In (20) the Physical Complexity, C, is defined for a population of sequences of length ℓ [3]. The most important question is what does the length ℓ equal if the population of sequences is of variable length? The issue is what ℓ represents, which is the maximum possible complexity for the population [3], and which we will call the *complexity potential* C_P. The maximum complexity in (20) occurs when the per-site entropies sum to zero, $\sum_{i=1}^{\ell} H(i) = 0$, as there is no randomness in the sites (all contain information), i.e. $C = \ell$ [3]. So, the *complexity potential* equals the length,

$$C_P = \ell, \tag{23}$$

provided the population S is of sufficient size for accurate calculations, as found in (22), i.e. $|S|$ is equal or greater than $|D|\ell$. For a population of variable length sequences, S_V, the complexity potential, C_{V_P}, cannot be equivalent to the length ℓ, because it does not exist. However, given the concept of minimum sample size from (22), there is a length for a population of variable length sequences, ℓ_V, between the minimum and maximum length, such that the number of per-site samples up to and including ℓ_V is sufficient for the per-site entropies to be calculated. So the *complexity potential* for a population of variable length sequences, C_{V_P}, will be equivalent to its *calculable* length,

$$C_{V_P} = \ell_V. \tag{24}$$

If ℓ_V where to be equal to the length of the longest individual(s) ℓ_{max} in a population of variable length sequences S_V, then the operational problem is that for some of the later sites, between one and ℓ_{max}, the sample size will be less than the population size $|S_V|$. So, having the length ℓ_V equalling the maximum length would be incorrect, as there would be an insufficient number of samples at the later sites, and therefore $\ell_V \not\equiv \ell_{max}$. So, the length for a population of variable length sequences, ℓ_V, is the highest value within the range of the minimum (one) and maximum length, $1 \leq \ell_V \leq \ell_{\text{max}}$, for which there are sufficient samples to calculate the entropy. A function which provides the sample size at a given site is required to specify the value of ℓ_V precisely,

$$sampleSize(i : site) : int, \tag{25}$$

where the output varies between 1 and the population size $|S_V|$ (inclusive). Therefore, the length of a population of variable length sequences, ℓ_V, is the highest value within the range of one and the maximum length for which the sample size is greater than or equal to the alphabet size multiplied by the length ℓ_V,

$$sampleSize(\ell_V) \geq |D|\ell_V \wedge sampleSize(\ell_V + 1) < |D|\ell_V, \tag{26}$$

where ℓ_V is the length for a population of variable length sequences, and ℓ_{max} is the maximum length in a population of variable length sequences, ℓ_V varies between $1 \leq \ell_V \leq \ell_{max}$, D is the alphabet and $|D| > 0$. This definition intrinsically includes a minimum size for populations of variable length sequences, $|D|\ell_V$, and therefore is the counterpart of (22), which is the minimum population size for same length populations.

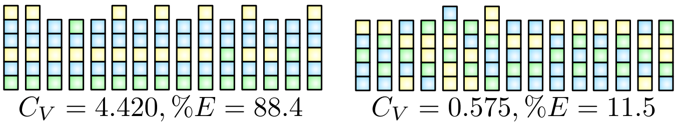

Figure 1: Abstract Visualisation for Populations of Variable Length Sequences: The Physical Complexity and Efficiency values are consistent with the intuitive understanding one would have for the complexity of the sample populations.

The length ℓ used in the limits of (14) no longer exists, and therefore (14) must be updated; so, the per-site entropy calculation for variable length sequences will be denoted by $H_V(i)$, and is,

$$H_V(i) = -\sum_{d \in D} p_d(i) \log_{|D|} p_d(i), \tag{27}$$

where D is still the alphabet, ℓ_V is the length for a population of variable length sequences, with the site i now ranging between $1 \leq i \leq \ell_V$, while the $p_d(i)$ probabilities still range between $0 \leq p_d(i) \leq 1$, and still sum to one. It remains algebraically almost identical to (14), but the conditions and constraints of its use will change, specifically ℓ is replaced by ℓ_V. Naturally, $H_V(i)$ ranges between zero and one, as did $H(i)$ in (14). So, when the entropy is maximum the character found in the site i is uniformly random, holding no information.

Therefore, the complexity for a population of variable length sequences, C_V, is the *complexity potential* of the population of variable length sequences minus the sum, over the length of the population of variable length sequences, of the per-site entropies (27),

$$C_V = \ell_V - \sum_{i=1}^{\ell_V} H_V(i), \tag{28}$$

where ℓ_V is the length for the population of variable length sequences, and $H_V(i)$ is the entropy for a site i in the population of variable length sequences.

Physical Complexity can now be applied to populations of variable length sequences, so we will consider the abstract example populations in Figure 1. We will let a single square, □, represent a site i in the sequences, with different colours to represent the different values. Therefore, a sequence of sites will be represented by a sequence of coloured squares, 🟨🟩🟦. Furthermore, the alphabet D is the set {🟨, 🟩, 🟦}, the maximum length ℓ_{max} is 6 and the length for populations of variable length sequences ℓ_V is calculated as 5 from (26). The Physical Complexity values in Figure 1 are consistent with the intuitive understanding one would have for the complexity of the sample populations; the population with high Physical Complexity has a little randomness, while the population with low Physical Complexity is almost entirely random.

4. Efficiency

Using our extended Physical Complexity we can construct a measure showing the use of the information space, called the Efficiency E, which is calculated by the Physical Complexity C_V over the complexity potential C_{V_P},

$$E = \frac{C_V}{C_{V_P}}. \tag{29}$$

The Efficiency E will range between zero and one, only reaching its maximum when the actual complexity C_V equals the complexity potential C_{V_P}, indicating that there is no randomness in the population. In Figure 1 the populations of sequences are shown with their respective Efficiency values as percentages, and the values are as one would expect.

The complexity C_V (28) is an absolute measure, whereas the Efficiency E (29) is a relative measure (based on the complexity C_V). So, the Efficiency E can be used to compare the complexity of populations, independent of their size, their length, and whether their lengths are variable or not (as it is equally applicable to the fixed length populations of the original Physical Complexity).

5. Clustering

The *complexity* of a population is the *clustering*, amassing of same or similar sequences, around the optimum genome [6]. The sequences of an evolving population will evolve, clustering around the optimal genome, assuming that its evolutionary process does not become trapped while clustering over local optima.

Clustering is indicated by the Efficiency E tending to its maximum, as the population's Physical Complexity C_V tends to the *complexity potential* C_{V_P}, because an optimal sequence is becoming dominant in the population, and therefore increasing the uniformity of the sites across the population. With a global optimum, the Efficiency E tends to a maximum of one, indicating that the *evolving population of sequences* is tending to a *set of clusters T of size one*,

$$E = \frac{C_V}{C_{V_P}} = 1 \; as \; |T| = 1, \tag{30}$$

assuming its evolutionary process does not become trapped at local optima. So, the Efficiency *E tending to its maximum* provides a *clustering coefficient*. It *tends to its maximum*, never quite reaching it, because of the mutation inherent in the evolutionary process.

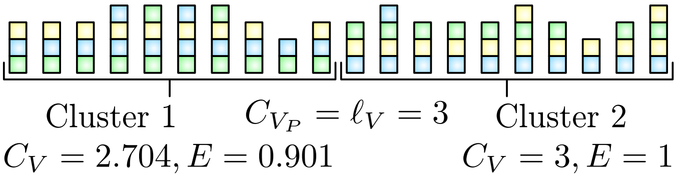

Figure 2: Population with Clusters Visible: Visualisation for a population of sequences with global optima, which has been arranged to show the clusters present.

If there are global optima the Efficiency E will tend to a maximum below one, because the population of sequences consists of more than one cluster, with each having an Efficiency tending to a maximum of one. The simplest scenario of clusters is *pure clusters*; *pure* meaning that each cluster uses a distinct (mutually exclusive) subset of the alphabet D relative to any other cluster. In this scenario the Efficiency E tends to a value based on the number of clusters $|T|$, because a *number* of the $p_d(i)$ probabilities at each *site* in (27) are the reciprocal of the number of clusters, $\frac{1}{|T|}$. So, given that the *number* of the $p_d(i)$ probabilities taking the value $\frac{1}{|T|}$ is equal to the number of clusters, while the other $p_d(i)$ probabilities take a value of zero, then the per-site entropy calculation of $H_V(i)$ from (27) becomes

$$H_V(i) = \log_{|D|} |T|, \tag{31}$$

where i is the site, $|D|$ is the alphabet size, and $|T|$ is the number of clusters. Hence, given (31), (28), and (24), then the Efficiency E from (29) becomes

$$E = 1 - (\log_{|D|} |T|), \tag{32}$$

where $|D|$ is the alphabet size and $|T|$ is the number of clusters. Therefore, the Efficiency E, the *clustering coefficient*, tends to a value that can be used to determine the number of *pure clusters* in an evolving population of sequences.

For a population S with clusters, each cluster is a sub-population with an Efficiency E tending to a maximum of one. To specify this relationship we require a function that provides the Efficiency E (29) of a population or sub-population of sequences,

$$\textit{efficiency(input :population) :int.} \tag{33}$$

So, for a population S consisting of a set of clusters T, each member (cluster) t is therefore a sub-population of the population S, and is defined as

$$t \in T = \left(t \subseteq S \wedge \textit{efficiency}(t) = 1 \wedge |t| \approx \frac{|S|}{|T|} \wedge \sum_{t \in T} |t| = |S|\right), \tag{34}$$

where a cluster t has an Efficiency E tending to a maximum of one, and the cluster size $|t|$ is approximately equal to the population size $|S|$ divided by the number of clusters $|T|$. It is only *approximately equal* because of variation from mutation, and because the population size may not divide to a whole number. These conditions are true for all members t of the set of clusters T, and therefore the summation of the cluster sizes $|t|$ equals the size of the population $|S|$.

A population of sequences with global optima, arranged to show the clustering, is shown in Figure 2. The clusters of the population have Efficiency values tending to a maximum of one, compared to the Efficiency of the population as a whole, which is tending to a maximum significantly below one.

The population size $|S|$, in Figure 2, is double the minimum requirement specified in (26), so that the complexity C_V (28) and Efficiency E (29) could be used in defining the principles of clustering without redefining the *length of a population of variable length sequences* ℓ_V (26). However, when determining the variable length ℓ_V of a cluster t, the sample size requirement is different, specifically a cluster t is a sub-population of S, and therefore by definition cannot have a population size equivalent to S (unless the population consists of only one cluster). Therefore, to manage clusters requires a reformulation of ℓ_V (26) to the highest value within the range of one and the maximum length for which the sample size is greater than or equal to the alphabet size multiplied by the length and divided by the number of clusters,

$$\ell_V = \begin{pmatrix} sampleSize(\ell_V) \approx \frac{|D|\ell_V}{|T|} \wedge \\ sampleSize(\ell_V + 1) < \frac{|D|\ell_V}{|T|} \end{pmatrix}, \tag{35}$$

where ℓ_{max} is the maximum length in a population of variable length sequences, ℓ_V varies between $1 \leq \ell_V \leq \ell_{max}$, D is the alphabet, $|D| > 0$, and T is the set of clusters in the population S.

A population with clusters will always have an Efficiency E tending towards a maximum significantly below one. Therefore, managing populations with clusters requires a reformulation of the Efficiency (29) to

$$E_c(S) = \begin{cases} \frac{C_V}{C_{V_P}} & \text{if } |T| = 1 \\ \frac{\sum\limits_{t \in T} E_c(t)}{|T|} & \text{if } |T| > 1 \end{cases}, \tag{36}$$

where t is a cluster, and a member of the set of clusters T of the population S. So, the Efficiency E_c is equivalent to the Efficiency E if the population consists of only one cluster, but if there are clusters then the Efficiency E_c is the average of the Efficiency E values of the clusters.

6. Simulation and Results

A simulated population S of sequences, $[s_1, s_1, s_2, ...]$, was evolved to a *selection pressure* R. A dynamic population size was used to ensure exploration of the available search space, which increased with the average length of the population. The optimal sequences were evolved to a *fitness function* generated from the *selection pressure* R. Each sequence of the population consisted of a list of sites, $[i_1, i_2, ...]$, while the *selection pressure* consisted of a list of attributes, $[r_1, r_2, ...]$. So, the *fitness function* for evaluating a sequence s, relative to the *selection pressure* R, was

$$fitness(s, R) = \frac{1}{1 + \sum_{r \in R} |r - i|}, \tag{37}$$

where i is a site of a sequence s measured against corresponding site r of the *selection pressure* R. Equation 37 was used to assign *fitness* values between 0.0 and 1.0 to each individual of the current generation of the population, directly affecting their ability to replicate into the next generation. The evolutionary computing process was encoded with a low mutation rate, a fixed selection pressure and a non-trapping fitness function (i.e. did not get trapped at local optima). The type of selection used was *fitness-proportional* and *non-elitist*, *fitness-proportional* meant that the *fitter* the individual the higher its probability of surviving to the next generation. *Non-elitist* meant that the best individual from one generation was not guaranteed to survive to the next generation; it had a high probability of surviving into the next generation, but it was not guaranteed as it might have been mutated. *Crossover* (recombination) was then applied to a randomly chosen 10% of the surviving population, a *one-point crossover*, by aligning two parent individuals and picking a random point along their length, and at that point exchanging their tails to create two offspring. *Mutations* were then applied to a randomly chosen 10% of the surviving population; *point mutations* were randomly located, consisting of *insertions*, *replacements*, and *deletions*. The issue of bloat was controlled by augmenting the *fitness function* with a *parsimony pressure* which biased the search to shorter sequences, evaluating longer than average sequences with a reduced *fitness*, and thereby providing a dynamic control limit which adapted to the average length of the ever-changing evolving populations.

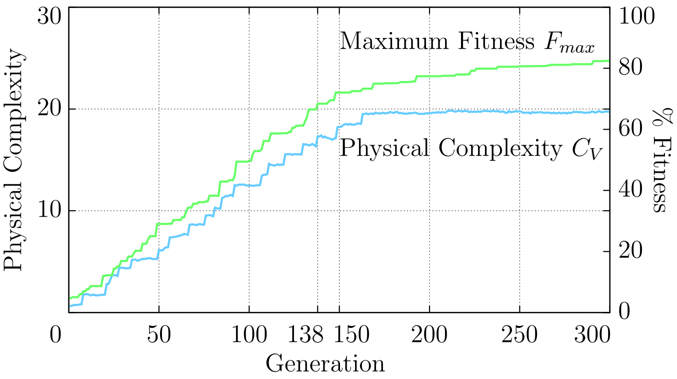

Figure 3: Graph of Physical Complexity and Maximum Fitness over the Generations: The Physical Complexity for variable length sequences increases over the generations, showing short-term decreases as expected, such as at generation 138.

Figure 3 shows, for a typical evolving population, the Physical Complexity C_V (28) for variable length sequences and the *maximum fitness* F_{max} over the generations. It shows that the fitness and our extended Physical Complexity; both increase over the generations, synchronised with one another, until generation 160 when the *maximum fitness* tapers off more slowly than the Physical Complexity. At this point the optimal length for the sequences is reached within the simulation, and so the advent of new fitter sequences (of the same of similar length) creates only minor fluctuations in the Physical Complexity, while having a more significant effect on the *maximum fitness*. The similarity of the graph in Figure 3 to the graphs in [4] confirms that the Physical Complexity measure has been successfully extended to variable length sequences.

6.1. *Efficiency*

Figure 4 is a visualisation of the simulation, showing two alternate populations that were run for a thousand generations, with the one on the left from Figure 3 run under normal

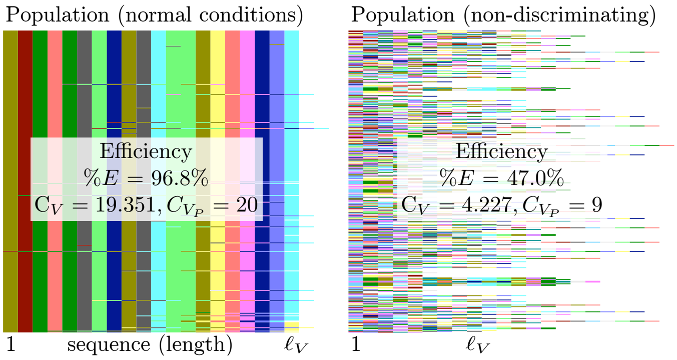

Figure 4: Visualisation of Evolving Populations at the 1000th Generation: The population on the left from Figure 3 was run under normal conditions, while the one on the right was run with a non-discriminating selection pressure.

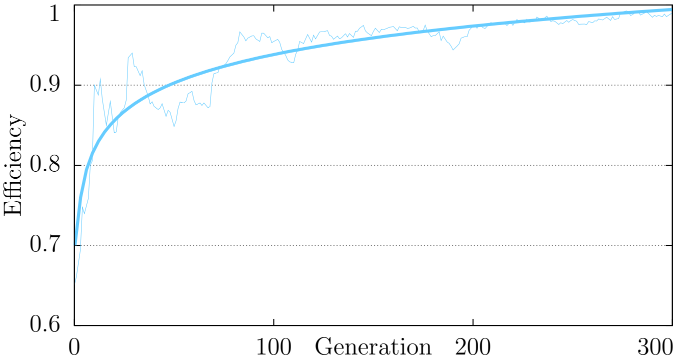

Figure 5: Graph of Population Efficiency over the Generations for the population from Figure 3: The Efficiency tends to a maximum of one, indicating that the population consists of one cluster.

conditions, while the one on the right was run with a non-discriminating selection pressure [18]. Each multi-coloured line represents a sequence, while each colour represents a site. The visualisation shows that our Efficiency E accurately measures the complexity of the two populations.

Figure 5 shows the Efficiency E (29), over the generations, for the population from Figure 3. The Efficiency tends to a maximum of one, indicating that the population consists of one cluster, which is confirmed by the visualisation of the population in Figure 4 (left).

6.2. *Clustering*

To further investigate the complexity of evolving populations, we simulated a typical population with a multi-objective *selection pressure* that had two independent global optima, and so the potential to support two *pure clusters* (each cluster using a unique subset of the alphabet D). The graph in Figure 6 shows the Efficiency E over the generations acting as a *clustering coefficient*, oscillating around the included best fit curve, quite significantly at the start, and then decreasing as the generations progressed. The Efficiency tended to 0.744, as expected from (32) given the alphabet size was fifteen, $|D|$=15, and the number of clusters was two, $|T|$=2. It indicated the occurrence of clustering, while the value it tended to indicated, as expected, the presence of two clusters in the population. A visualisation of the population is shown in Figure 7, in which the sequences were grouped to show the two clusters. As expected from (35) each cluster had a much higher Physical Complexity and Efficiency compared to the population as a whole. However, the Efficiency E_c is immune to the clusters and therefore calculated the complexity of the population correctly.

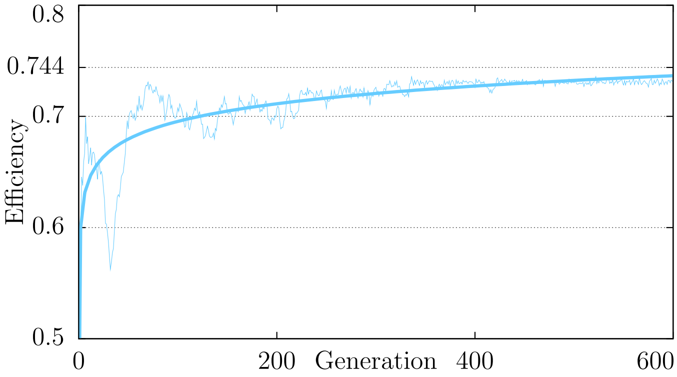

Figure 6: Graph of the Clustering Coefficient over the Generations: The Efficiency oscillated to 0.744, as expected from (32) given the alphabet size was fifteen, $|D|$=15, and the number of clusters was two, $|T|$=2, indicating more than one cluster.

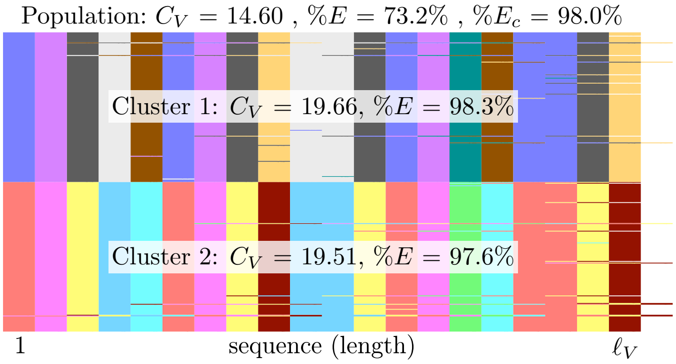

Figure 7: Visualisation of Clusters in an Evolving Population at the 1000th Generation: The sequences were grouped to show the two clusters.

7. Conclusions

Physical Complexity [4] is based upon the individuals of a population within the context of their environment, but was only formulated for populations of fixed length. However, this was not a fundamental property of its definition [4], and so we chose to extend it to include populations of variable length sequences. We then built upon this to construct a variant of the Physical Complexity called the Efficiency, because it was based on the efficiency of information storage, which was then used to develop an understanding of clustering and atomicity in evolving populations with multi-objective *selection pressures*. The *clustering coefficient* defined by the Efficiency tending to its maximum, not only indicates clustering, but can also determine the number of clusters (for *pure clusters*).

We then investigated our extended Physical Complexity through experimental simulations, with the results consistent with the original, confirming a successful algebraic reformulation to include populations of variable length sequences.

We then investigated the Efficiency, which performed as expected, confirmed by the numerical results and the visualisations matching our intuitive understanding. We then applied the Efficiency to the determination of clusters for populations with multi-objective *selection pressures*. The numerical results, combined with the visualisations of multi-cluster populations, confirmed the ability of the Efficiency to act as a *clustering coefficient*, not only indicating the occurrence of clustering, but also the number of clusters. Finally, we confirmed that the *Efficiency* E_c (for populations with clusters) could calculate correctly the Physical Complexity for variable length populations with clusters.

Overall, Physical Complexity has been extended to include populations of variable length, with our Efficiency definition providing a macroscopic value to characterise the level of Physical Complexity. Furthermore, our Efficiency E_c provides a normalised *universally applicable* macroscopic value to characterise the complexity of any population, independent of clustering, atomicity, length (variable or same), and size. Most importantly, the understanding and techniques we have developed have applicability as wide as the original Physical Complexity, which has been applied from DNA [3] to simulations of self-replicating programmes [21].

8. Acknowledgments

The authors would like to thank the following for encouragement and suggestions; Dr Paolo Dini of the London School of Economics and Political Science, and Dr Christoph Adami of Keck Graduate Institute.

References

[1] C. Adami. *Introduction To Artificial Life*. Springer, 1998.

[2] C. Adami. Sequence complexity in darwinian evolution. *Complexity*, 8:49–56, 2003.

[3] C. Adami and N. Cerf. Physical complexity of symbolic sequences. *Physica D*, 137:62–69, 2000.

[4] C. Adami, C. Ofria, and T. Collier. Evolution of biological complexity. *Proceedings of the National Academy of Sciences*, 97:4463–4468, 2000.

[5] G. Basharin. On a statistical estimate for the entropy of a sequence of independent random variables. *Theory Probability and its Applications*, 4:333–336, 1959.

[6] M. Begon, J. Harper, and C. Townsend. *Ecology: Individuals, Populations and Communities*. Blackwell Publishing, 1996.

[7] G. Briscoe. *Digital Ecosystems*. PhD thesis, Imperial College London, 2009.

[8] G. Briscoe and P. De Wilde. Digital Ecosystems: Evolving service-oriented architectures. In *IEEE Bio Inspired Models of Network, Information and Computing Systems Conference*, 2006.

[9] G. Briscoe and P. De Wilde. Computing of applied digital ecosystems. In *ACM Management of Emergent Digital Ecosystems Conference*, 2009.

[10] G. Briscoe and P. De Wilde. Digital Ecosystems: Self-organisation of evolving agent populations. In *ACM Management of Emergent Digital Ecosystems Conference*, 2009.

[11] G. Briscoe and P. De Wilde. The computing of digital ecosystems. *International Journal of Organizational and Collective Intelligence*, 1(4), 2010.

[12] G. Briscoe and S. Sadedin. Natural science paradigms. In *Digital Business Ecosystems*, pages 48–55. European Commission, 2007.

[13] G. Briscoe, S. Sadedin, and G. Paperin. Biology of applied digital ecosystems. In *IEEE Digital Ecosystems and Technologies Conference*, pages 458–463, 2007.

[14] M. Gell-Mann and S. Lloyd. Effective complexity. In M. Gell-Mann and C. Tsallis, editors, *Nonextensive Entropy: Interdisciplinary Applications*, pages 387–389. Oxford Univ Press, 2003.

[15] T. Gregory. Coincidence, coevolution, or causation? DNA content, cellsize, and the C-value enigma. *Biological Reviews*, 76:65–101, 2001.

[16] I. Grosse, P. Bernaola-Galván, P. Carpena, R. Román-Roldán, J. Oliver, and H. Stanley. Analysis of symbolic sequences using the Jensen-Shannon divergence. *Physical Review E*, 65(4):041905, 2002.

[17] E. Jaynes. *Probability theory: the logic of science*. Cambridge University Press, 2003.

[18] M. Kimura. *The neutral theory of molecular evolution.* Cambridge University Press, 1983.

[19] A. Kolmogorov. Three approaches to the definition of the concept "quantity of information". *Problems of Information Transmission*, 1:3–11, 1965.

[20] P. Lamberti and A. Majtey. Non-logarithmic Jensen-Shannon divergence. *Physica A: Statistical Mechanics and its Applications*, 329(1-2):81–90, 2003.

[21] E. Lenski, C. Ofria, R. Pennock, and C. Adami. The evolutionary origin of complex features. *Nature*, 423:139–144, 2003.

[22] M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, 1997.

[23] S. Lloyd and H. Pagels. Complexity as thermodynamic depth 1. *Annals of Physics*, 188(1):186–213, 1988.

[24] D. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[25] M. Martin, F. Pennini, and A. Plastino. Fisher's information and the analysis of complex signals. *Physics Letters A*, 256(2-3):173–180, 1999.

[26] D. Queirós and M. Sílvio. On a comparative study between dependence scales determined by linear and non-linear measures. *Physica D: Nonlinear Phenomena*, 238(7):764–770, 2009.

[27] S. Queirós. On non-Gaussianity and dependence in financial time series: a nonextensive approach. *Quantitative Finance*, 5(5):475–487, 2005.

[28] C. Thomas Jr. The genetic organization of chromosomes. *Annual Reviews in Genetics*, 5:237–256, 1971.

[29] C. Tsallis, C. Anteneodo, L. Borland, and R. Osorio. Nonextensive statistical mechanics and economics. *Physica A: Statistical Mechanics and its Applications*, 324(1-2):89–100, 2003.

[30] L. von Bertalanffy. *General system theory: Foundations, Development, Applications*. Braziller, 1973.

[31] W. Zurek. Algorithmic information content, church-turing thesis. physical entropy, and maxwll's demon. In W. Zurek, editor, *Complexity, Entropy, and the Physics of Information*, pages 73–89. Addison-Wesley, 1990.

[32] A. Zvonkin and L. Levin. The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. *Russian Mathematical Surveys*, 25:83–124, 1970.